----------------------- CEN 4072/6070 Software Testing & Verification ---------------------

Exam 2 – Spring 2013 – Solution Notes

1. a. would not, b. would, c. would not, d. would not, e. would, f. would

2. a. true, b. false, c. true, d. false, e. true, f. true, g. false, h. true, i. false, j. true

3. a.

$$\frac{sp(S,P) \Rightarrow Q}{\{P\}\ S\ \{Q\}}$$

b. $sp(\text{if } b \text{ then } S, P) \equiv sp(S, b \wedge P)) \vee (\neg b \wedge P)$

c. By the sp ROI, we need to show: $sp(\text{if } b \text{ then } S, P) \Rightarrow Q$

$$\begin{aligned} sp(\text{if } y>0 \text{ then } y := y - (2*y), y=x) &= sp(y := y - (2*y), y>0 \wedge y=x) \vee (y\le 0 \wedge y=x) \\ &= (y=y'-2y' \wedge y'>0 \wedge y'=x) \vee (y\le 0 \wedge y=x) \\ &= (y=-x \wedge x>0 \wedge y'=x) \vee (y\le 0 \wedge y=x) \end{aligned}$$

To prove $[\ (y=-x \wedge x>0 \wedge y'=x) \vee (y\le 0 \wedge y=x)\ ] \Rightarrow Q$, it must be shown that:

(1) $(y=-x \wedge x>0 \wedge y'=x) \Rightarrow Q$ **and** that
(2) $(y\le 0 \wedge y=x) \Rightarrow Q$.

For (1), we have: $(y=-x \wedge x>0 \wedge y'=x) \Rightarrow (y=-x \wedge y'=x) \Rightarrow Q$
and for (2) we have: $(y\le 0 \wedge y=x) \Rightarrow y=x \Rightarrow Q$.

4.

	f_1	f_2	f_3	f_4
P	S	N	C	N

5. function of while statement: $(x\ge 5 \rightarrow x,y := 5,y)$
function of if then statement: $(y>0 \rightarrow (x\ge 5 \rightarrow x,y := 5,y) \mid y\le 0 \rightarrow x,y := x,y)$
function of initial assignment statement: $(x,y := x,x-2)$

Therefore, the function of the compound program is...

$= (y>0 \rightarrow (x\ge 5 \rightarrow x,y := 5,y) \mid y\le 0 \rightarrow x,y := x,y)\ o\ (x,y := x,\mathbf{x-2})$
$= (\mathbf{x-2}>0 \rightarrow (x\ge 5 \rightarrow x,y := 5,\mathbf{x-2}) \mid \mathbf{x-2}\le 0 \rightarrow x,y := x,\mathbf{x-2})$
$= (\mathbf{x}>2 \rightarrow (x\ge 5 \rightarrow x,y := 5,\mathbf{x-2}) \mid \mathbf{x}\le 2 \rightarrow x,y := x,\mathbf{x-2})$

$= \mathbf{(x\ge 5 \rightarrow x,y := 5,x-2 \mid x\le 2 \rightarrow x,y := x, x-2)}$

6. a. valid, b. invalid, c. valid, d. invalid, e. invalid, f. invalid, g. valid, h. valid, i. valid, j. invalid

7. a. Theorem.

Let f = [while p do g]. If $X_0 \in D(f)$, $X \in D(f)$, and $q(X) = (f(X)=f(X_0))$, then q is an invariant of while p do g; i.e., it has the following properties:

1. $q(X_0)$ is true, and
2. $(q(X) \wedge p(X)) \Rightarrow q\text{o}g(X)$.

In addition, q(X) is an f-adequate invariant; i.e.,

3. $(q(X) \wedge \neg p(X)) \Rightarrow (X=f(X_0))$.

b. q = $\mathbf{(p = p_0 2^{k_0-k} \wedge k \geq 0)}$ (Setting $t(X)=t(X_0)$ for variables p, k yields: $[0=0 \wedge p2^{\mathbf{k}} = p_0 2^{\mathbf{k_0}}] \Rightarrow \mathbf{p = p_0 2^{k_0-k}}$)

c. Property (2): Does $(q(X) \wedge p(X))$ => $q\text{o}g(X)$?

$(q(X) \wedge p(X)) = (p = p_0 2^{\mathbf{k_0-k}} \wedge k \geq 0 \wedge k \neq 0)$

Does $(p = p_0 2^{\mathbf{k_0-k}} \wedge k \neq 0)$ => $((p = p_0 2^{\mathbf{k_0-k}}) \text{ o } (p,k := \mathbf{2p},\mathbf{k-1}))$ **?**

$$= (\mathbf{2p} = p_0 2^{k_0-(\mathbf{k-1})})$$
$$= (p = p_0 2^{\mathbf{k_0-(k-1)-1}})$$
$$= (p = p_0 2^{\mathbf{k_0-k}})) \quad \checkmark$$

Therefore, $(q(X) \wedge p(X))$ => $q\text{o}g(X)$ ✓

Property (3): Does $(q(X) \wedge \neg p(X)) \Rightarrow (X=f(X_0))$?

$$(q(X) \wedge \neg p(X)) = (p = p_0 2^{\mathbf{k_0-k}} \wedge k \geq 0 \wedge k=0) \Rightarrow (p = p_0 2^{\mathbf{k_0}} \wedge k=0)$$
$$\Rightarrow (p = t_p(p_0) = p_0 2^{\mathbf{k_0}} \wedge k = t_k(k_0) = 0)$$
$$\Rightarrow (X=t(X_0)). \quad \checkmark$$

d. **Does ¬*p* => (*t* = I) ?**

$$(k=0) \text{ => } (t = (p,k := p2^{\mathbf{0}},0) = (p,k := p,k) = I) \quad \checkmark$$

7. d. (cont.)

Does *p* => (*t* = *t* o *g*) ?

There are 2 cases to consider: k<0 and k>0.

case a:

(k<0) => (*t* = ***undefined***) √

(k<0) => ((*t* o *g*) = undefined o *g*

(since ((k<0) o g(k<0)) = *true)*

= ***undefined***) √

case b:

(k>0) => (*t* = **(p,k := p2^{k},0)**) √

(k>0) => ((*t* o *g*) = (p,k := p2^{k},0) o (p,k := 2p,k-1)

(since ((k≥0) o g(k>0)) = *true)*

= (p,k := (2p)2$^{(k-1)}$,0)

= **(p,k := p2^{k},0)**) √

Therefore, *p* => (*t* = *t* o *g*)

Therefore, *t* = [*T*], assuming term*(t,T).*

e. H_0: ¬b ∧ Q ≡ k=0 ∧ p=16

H_1: b ∧ wp(S, H_0) ≡ k≠0 ∧ wp(p := p∗2; k := k-1, k=0 ∧ p=16)
≡ k≠0 ∧ **k-1**=0 ∧ **2p** = 16
≡ k=1 ∧ p=8

H_2: b ∧ wp(S, H_1) ≡ k≠0 ∧ wp(p := p∗2; k := k-1, k=1 ∧ p=8)
≡ k≠0 ∧ **k-1**=1 ∧ **2p**=8)
≡ k=2 ∧ p=4

$H_{n>0}$: b ∧ wp(S, H_{n-1}) ≡ **k=n ∧ p=2$^{(4-n)}$** ≡ **k=n ∧ p=16/2^{n}**)

f. wp in closed form: (k=0 ∧ p=16) V (0<k≤4 ∧ p=16/2^{k}) (Note that (k>4 ∧ p=16/2^{k}) is FALSE since p can only assume INTEGER values.)

= **(0≤k≤4 ∧ p=2$^{(4-k)}$)** = **(0≤k≤4 ∧ p=16/2^{k})**

Equivalently, this could be written as:

[(k=0 **∧** p=16) V (k=1 **∧** p=8) V (k=2 **∧** p=4) V (k=3 **∧** p=2) V (k=4 **∧** p=1)]

7. g. wlp(T,Q) = wp(T,Q) ∨ ¬wp(T,true)

wlp in closed form: $\mathbf{(0 \le k \le 4 \wedge p=2^{(4-k)}) \vee k<0 = (0 \le k \le 4 \wedge p=16/2^{k}) \vee k<0}$

(Note that T does not terminate for k<0.)

8. **INITIALIZATION: Does P ⇒ I?**

P: (p=1 ∧ k=n) ⇒ ($p=\mathbf{2^{n-k}}$ = $1=2^{n-n}$ = $1=2^0$ = 1=1 = true)
Therefore P ⇒ I. √

PRESERVATION: Does {I ∧ b} s {I}?

I ∧ b: {$p=2^{\mathbf{n-k}}$ ∧ k≠0}
p := p∗2
{ $p=2^{\mathbf{n-k+1}}$ ∧ k≠0 }
k := k−1
{ $p=2^{\mathbf{n-(k+1)+1}}$ ∧ k+1≠0 }
⇔
{ $p=2^{\mathbf{n-k}}$ ∧ k+1≠0 } ⇒ I √

FINALIZATION: Does (I ∧ ~b) ⇒ Q?

(I ∧ ¬b): ($p=2^{\mathbf{n-k}}$ ∧ k=0) ⇒ $p=2^{\mathbf{n}}$ = Q √

9. a. false, b. true, c. false, d. true, e. true

10. a. true, b. false, c. false, d. false, e. true

11. a. i.
b. A fundamental property of all functions, f, computed by a while loop is that q(X) = (f(X) = $f(X_0)$) for all execution states, X, that hold when the loop predicate is evaluated. This implies that the final state, $X_n = f(X_0) = f(X_n)$.

Let $X_0 = (x_0, y_0)$ = (a,b). Then for the function, f, given above, we have

$f(X_0) = X_n$ = (a,a+b) and $f(X_n)$ = (a,a+b+a)

Thus, $f(X_0) \neq f(X_n)$ (except for the special case $x_0=0$), and we therefore conclude that there is no program of the form "while b do s" that computes function f for all X∈D(f).

(Equivalently, we could show that while $\boldsymbol{q}$**(X)** ⇔ **($x=x_0 \wedge y+x=y_0+x_0$)** agrees with **X_0, it does <u>not</u> agree with X_n**.)